CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A,
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: (21) 05-561325
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

91-04080/

CRT-762

An Interactive Graphic Approach for the Integrated Design of Intercity Transportation Timetables and Vehicle Operations

by
Christian Lardinois[1]
Teodor Gabriel Crainic[2]
Johanne Dion[1]

[1] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale A, Montréal, Canada H3C 3J7.

[2] Centre de recherche sur les transports, Université de Montréal and Département des sciences administratives, Université du Québec à Montréal, C.P. 6192, succursale A, Montréal, Canada H3C 4R2.

Centre de recherche sur les transports — Publication #762

March 1991

Dépôt légal – 1er trimestre 1991

ABSTRACT

The planning and management of the services offered by any intercity passenger transportation company involve two key levels of concern that can be characterized as follows: (1) the strategic "market-oriented" level where one is mainly concerned with the design of an efficient timetable that best suits travellers' expectations, but with little consideration for operational and feasibility issues, and (2) the operational level where, on the contrary, one mainly addresses such issues – notably the routing and scheduling of vehicles (which ultimately determines the service timetable) –, to the detriment of marketing considerations. In this paper, a decision support system is presented which aims at reconciling both levels of concern within an integrated interactive-graphic framework. The main components of the computer system are described, its database structure and the underlying algorithmic method are outlined, and experimental results are reported.

Keywords: Intercity passenger transportation; Decision support system; Service timetable; Vehicle routes and schedules.

RÉSUMÉ

La planification et la gestion des services offerts par toute compagnie de transport interurbain de passagers impliquent deux niveaux de préoccupation distincts qu'on peut caractériser comme suit: (1) le niveau stratégique, axé sur des considérations de type "marketing", où l'on s'adresse essentiellement au design d'un horaire de service efficace qui satisfait autant que possible les attentes des voyageurs, mais sans trop se préoccuper des questions d'opération et de faisabilité, et (2) le niveau opérationnel où, au contraire, on s'adresse essentiellement à ces questions d'opération et de faisabilité, au détriment des objectifs de marketing; la conception des routes et horaires des véhicules (qui en fait déterminent l'horaire de service qui sera annoncé aux voyageurs) constitue le problème essentiel à ce second niveau. Dans cet article, on présente un système interactif-graphique d'aide à la décision qui intègre les deux niveaux de préoccupation. Les principales composantes du système sont décrites, la structure de sa banque de données et les algorithmes qui ont été développés sont présentés dans leurs grandes lignes, et on fait état de quelques résultats expérimentaux.

Mots-clés: Transport interurbain de voyageurs; Système d'aide à la décision; Horaire annoncé aux voyageurs; Routes-horaires des véhicules.

1. INTRODUCTION AND PROBLEM-SETTING

For any intercity passenger transportation company, the construction of timetables for travellers can be a very complex exercise. However, in the existing literature on the supply side of intercity passenger transportation systems [1], this problem is rarely dealt with directly. In fact, most schedule-related papers in transportation focus either on normative models for the routing and scheduling of vehicles and crews (see [2] among many others), or on the evaluation of given vehicle schedules with passenger assignment models (see for example [3] in the air transportation sector). Little or nothing is said about the construction of the corresponding service timetable intended for travellers, although any company should coordinate the vehicle routing and scheduling with the design of the timetable.

Of course, many companies have developed their own in-house systems or procedures to modify their timetable from time to time, but more often than not they confine themselves to marginal modifications. Many of these systems are manual, while others, though completely or partially computerized, are rather heavy to operate. Thus, timetable construction is still a complex issue, in particular when a deregulated and increasingly competitive environment requires drastic changes in operating schedules or for totally new ones. This is the case, for instance, when an airline replaces direct point-to-point services with hub-and-spoke operations. In addition, for marketing reasons, the timetable issue cannot be overlooked: whatever the actual routes and schedules of buses, trains and airplanes, travellers and travel agencies have no choice but to rely on available information, i.e. the service offerings

publicized in official timetable guides, whether printed in commercial publications or displayed on computer terminals.

In its simplest form, the intercity passenger transportation timetabling problem can be stated as follows. Given a set of vehicle routes and schedules, determine all travel alternatives (itineraries and schedules) to be published in all markets (city pairs), including the indirect alternatives, i.e. those with stops or connections. Even considered from this angle, the task is not easy. All direct (nonstop) travel alternatives are easily identified and, of course, they are all published, but the indirect ones involve complex analysis and difficult choices. On the one hand, publishing all travel schedules along all possible indirect itineraires will result in an unreadable timetable with multiple uninteresting and unrealistic itineraries and schedules. On the other hand, publishing only the most plausible or natural indirect travel alternatives – with the best connections only – will lead to a "poor" timetable, unrepresentative of all other alternatives that travellers could actually use (e.g., as second choices when congestion occurs, or for any other reason). A proper balance must be established between the quantity of information to be released in the timetable, its relevance to different traveller attitudes, and its usefulness in reassigning travellers who have missed a connection or those in excess of the capacity of the services normally used.

The discussion above supposes that vehicle operations are given and fixed. This is not the case in the real world, and accordingly one of the main necessary features of an efficient timetabling procedure is to update the timetable quickly and appropriately in response to changes in the vehicle routes or schedules. Such changes can be

intentional, for instance when they precede an anticipated variation in transportation demand. They may also occur accidentally, as is the case when a vehicle develops mechanical problems, or for any other reason that results in delaying or cancelling some services. Any change in scheduled vehicle operations between two cities (especially when this is unforeseen) directly affects travellers in that market, as well as those in other markets for which these two cities are or could be intermediate stops or connecting points. Some travel alternatives and connections currently in the timetable must be modified or cancelled, while others which are not included in the timetable become possible or more attractive and should be announced.

As noted above, the most complex situation occurs when the company has considerable freedom of action and engages in a drastic revision of most or all of its operations and services (possibly leaving some markets and entering new ones). Such a situation can be seen from two different angles, corresponding to two different levels of concern in the company, with different and, in fact, almost conflicting objectives. At the first level, the company is mainly concerned with *strategic issues and marketing objectives* regarding the level and quality of service to be supplied in every market. The focus is on the design of a timetable which best suits travellers' expectations and which will effectively compete with the services offered by other carriers; a myopic view at this level will tend to promote the maximization of the company's output, regardless of cost. At the other level, the company is mainly concerned with *operational issues*, and in particular with the design of vehicle routes and schedules which will ultimately form the actual service timetable. At this level, one aims

at minimizing costs while taking into account all operating factors and constraints (vehicle fleet size, crew scheduling, maintenance requirements, union rules, etc.); a myopic view here will focus on operating costs and feasibility issues, to the detriment of the level and quality of service. Thus, the organisation is faced with two different perspectives that must be reconciled within the planning/management process.

In this paper, we present a user-friendly interactive graphic approach which has been implemented on a personal computer to assist in establishing the vehicle routes and schedules and the service timetable to be announced to travellers, while integrating both the above-mentioned levels of concern.

Note that we do not address here the problem of the optimal design of vehicle routes and schedules, and we do not attempt to identify the best timetable with respect to marketing or other issues; the users of the system are assumed to have their own decision-making criteria and, possibly, their own tools for this. However, we do propose a simple but effective decision support system for integrated timetabling and vehicle fleet management.

The following section presents an overview of the system. Next, we describe in some detail each of its main components. Then, we outline the database structure, along with a few words on tests with the system. Conclusions are presented in the final section of the paper.

2. OVERVIEW OF THE SYSTEM

In accordance with our previous identification of two different but interdependent levels of concern (strategic vs. operational), the system we have developed comes in the form of an iterative decision-making process subdivided into two levels. At both levels, analysis and decision making take place on the display screen in an interactive and partly graphic mode, and one can move freely from one level to the other at any time. A flow chart presenting the four components of the overall system is shown in Fig. 1.

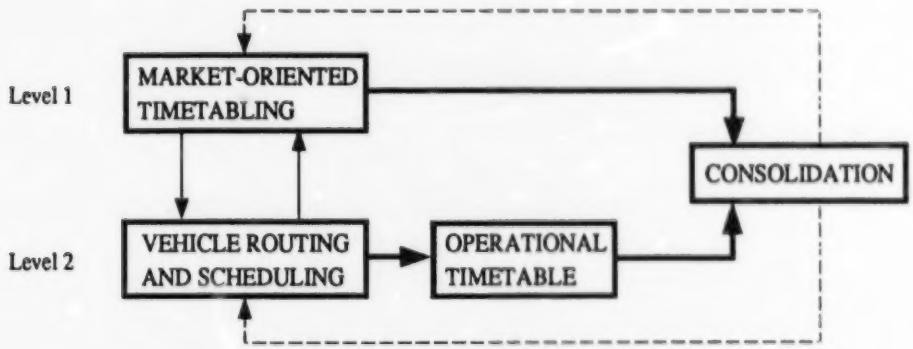

Fig. 1. Components and structure of the system.

The *Market-Oriented Timetabling Module* is used to draft the timetable as it is conceived at the strategic level. This task, which is the concern of the marketing department in the company, includes the choice of markets to be serviced and the preliminary design of the service schedules to be published in each market. At this level, no attention is paid to vehicle operations, at least during the first iteration. The "market-oriented timetable" produced in this way can be regarded as a target.

The *Vehicle Routing and Scheduling Module* and the *Operational Timetable Module* correspond to the operational level, or operations department. The first of these two modules is used to establish or modify the vehicle routes and schedules, while the second automatically generates the "operational timetable" resulting from these vehicle operations as soon as they are defined or modified. In the course of this process, the user of the system tries to arrange the vehicle operations so as to make the resulting operational timetable as close as possible to the market-oriented timetable established at the first level.

The *Consolidation Module* is used to assist in comparing the timetables displayed at both levels, at any time during the decision-making process. On the basis of this comparison, one can go back to one or the other level and, accordingly, complete or modify the market-oriented timetable (level 1) or the vehicle operations and associated operational timetable (level 2). This process stops when the two timetables match satisfactorily, that is, when the marketing department agrees with the last vehicle operating plan and corresponding timetable proposed by the operations department.

Note that the Market-Oriented Timetabling Module (level 1) could be removed from the system by bringing the marketing issues directly into play at level 2, along with all operational issues. However, the advantage of the two-level approach is to provide the ability to deal explicitly with both levels of concern separately, to switch from one to the other coherently, and to reconcile them progressively within one single computerized framework in which different parts of the company can participate.

3. SYSTEM MODULES AND USER INTERFACES

The overall system is essentially a decision support system including interactive and partly graphical procedures for entering, modifying and displaying the data, as well as automatic computational procedures which perform all data-processing tasks. This includes the sorting, selection and assembly of the elements of schedule information to be released in the timetables at both levels of the system.

3.1 Level 1.

The user interface of the *Market-Oriented Timetabling Module* is illustrated in Fig. 2. The lower part of the computer screen is the workspace, i.e. the section where the user can handle the relevant data. The upper part, used only for consultation, is explained at the end of this section.

The workspace is divided into four windows, each corresponding to a city pair (specified by the user in the two upper corners of the window), and one can scroll the data in each window separately. Each of the first three windows is used to define or modify the direct (nonstop) services between any city pair, by stating their departure times along with the specification of the type of each such service (speed, capacity, etc.); arrival times are computed and displayed automatically by the system. As indicated previously, this task is performed without worrying about the actual vehicle routes corresponding to these services (at least during the first iteration).

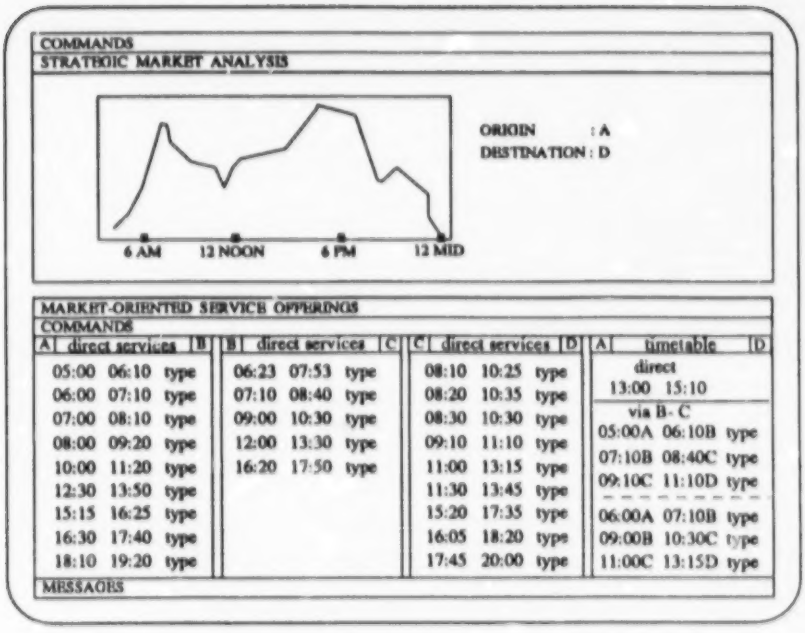

Fig. 2. Main user interface at level 1.

The fourth window is used to display the "market-oriented timetable", i.e. the tentative list of all service offerings, or travel alternatives, market by market. This timetable is automatically generated by the system from the direct services, and it is instantly updated in the database and on the display screen as soon as any direct service is entered, modified or cancelled in one of the first three windows. Each indirect service is displayed as a two- or three-line entry, which means that it is made up of two or three direct services. Of course, each direct travel alternative also is displayed in the fourth window, on a single line.

With this approach, each travel alternative published in the timetable passes through at most two intermediate cities. For each market, all these plausible in-

termediate cities are usually known or can be identified. They are listed a priori and included in the database, but the user can specify other itineraries provided they do not involve more than two intermediate cities. Travel itineraries with more transit points can be contemplated by considering combinations of published service offerings. Note that it would not be too difficult to modify the system to provide this information more systematically if desired (see last section of the paper).

A single window would have been enough to handle all direct services between all city pairs. The display screen exhibits three such windows in order to assist the user in visualizing the possible pairings of these direct services and their relationship with the resulting timetable. For example, using the first three windows to define or modify the direct services from city A to city B, then from B to C, and then from C to D, the user can consult and properly adjust the corresponding service offerings in the fourth window, notably the indirect services A-B-C, B-C-D and A-B-C-D in markets AC, BD and AD.

Given that each indirect service consists of direct services, a *pairing algorithm* has been developed to build all relevant pairings of direct services in every city. The space-time diagram in Fig. 3 illustrates the pairings of direct services from city X to city Y with direct services from city Y to city Z, together with the effect of the insertion of a new service between Y and Z. These pairings correspond to lines of the same type (e.g. continuous, dotted, ...) on both sides of Y. The central idea, formally defined in Appendix 1, is based on the concept of *coupled time-segments*, that is, on the pairing of the subgroups of incoming services *(arrival time-segments)*

and outgoing services *(departure time-segments)* which respectively have the same outgoing service as best successor and the same incoming service as best predecessor, each *arrival time-segment* being coupled with the *departure time-segment* which includes its best successor, and vice versa. This approach, which also takes account of constraints on the minimum and maximum stopping and connecting times at the intermediate city, does not include any pairings between services not belonging to coupled time-segments because such pairings are obviously less attractive than those between services belonging to coupled time-segments. However, it generates more service pairings than strictly the best ones, i.e. those only between the best predecessor and best successor within each pair of coupled time-segments. (Furthermore, the user could be given the power to select and force additional pairings besides those automatically generated by the system.) Note that, as indicated previously, the approach actually takes the form of an updating process, i.e. a process which updates all current pairings as soon as any direct service is created, modified or cancelled. Appendix 1 gives further details.

In Fig. 3, before the insertion of a new service, the first three incoming services form an arrival time-segment {1,2,3}, with service 5 as their best successor. Each of these three services is paired with the services {5,6,7}, which form a departure time-segment, with service 3 as their best predecessor. Service 7 is included in this time-segment because of the minimum stopping/connecting time constraint which prevents services 4 and 7 from making a pair. Service 4 forms an arrival time-segment by itself and is paired with the departure time-segment {8,9}. The insertion of service 10

between Y and Z (right part of the illustration) breaks the first arrival time-segment {1,2,3} into two new segments, namely {1,2} and {3}, which are paired with {10} and {5,6,7} respectively.

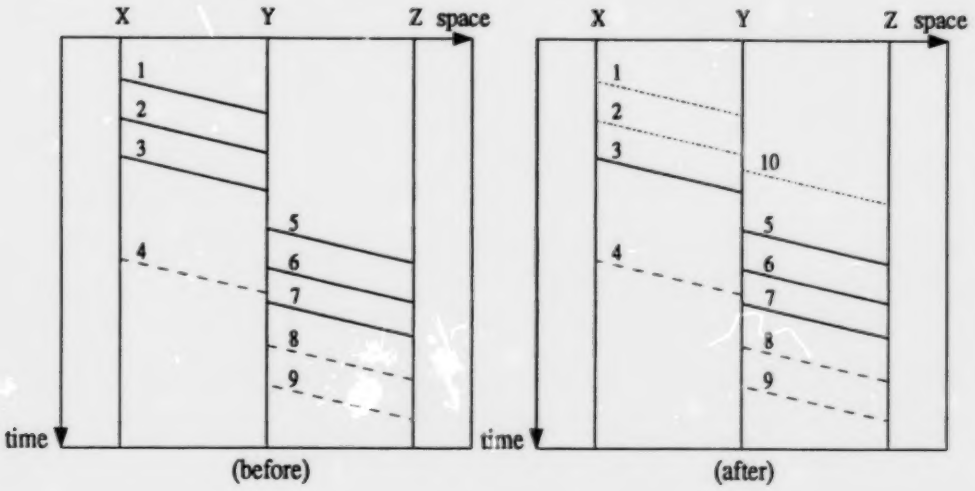

Fig. 3. Pairing of direct services.

Since the graphical representation in Fig. 3 is quite informative with respect to the possible pairing of direct services in space and time, it has been provided as a secondary user interface. This allows the user to easily define or modify all service offerings in an interactive-graphic mode, by entering and handling the appropriate data with the computer mouse.

The upper half of the first user interface (Fig. 2) is used to display any available and computerized information which may be of some help at this level of decision making, such as the results of periodic strategic market analysis. To illustrate this, a typical observed time-of-day variation of the transportation demand for a given

origin-destination pair is shown, although this has not been implemented so far. For the present, this screen section can be used for consultation of the "operational service offerings" established at level 2 of the system.

3.2 Level 2.

The user interface relative to level 2 is illustrated in Fig. 4. Again, the lower half of the display screen is the workspace, and the upper half is a consulting space. The former corresponds to the *Vehicle Routing and Scheduling Module*, while the latter here exhibits the output of the *Operational Timetable Module* (it is also used to display the output of the consolidation module; see section 3.3).

The vertical axis of the workspace is used to identify the vehicles; the horizontal axis represents the time scale. Each jagged line represents a vehicle route and schedule, with each bump corresponding to a movement of the vehicle between a city pair (identified below the line), and each hole corresponding to a stop in a city.

All these vehicle operations are defined or modified with the aid of mouse-supported commands and pop-up windows which allow different basic interactive-graphic tasks such as inserting a new vehicle route, moving a portion of an existing route, exchanging portions of two routes, defining the exact departure time from a city, scrolling everything up or down, etc.

The four windows in the upper half of the screen have the same meaning as those in Fig. 2, along with similar scrolling capabilities, except that here the schedule information which can be consulted in these windows results from the vehicle

operations entered in the lower half of the screen: as soon as any vehicle route and schedule is entered, modified or cancelled, the system immediately updates the list of direct services (notably those currently displayed in the first three windows), which in turn automatically and instantly leads to the updating of the "operational timetable" using the same pairing algorithm as at level 1; this timetable can be consulted in the fourth window. Note that in the present case, the information relative to the type of each service is replaced by the identification number of the vehicle route and schedule (#vrs in the illustration) to which the service belongs. Therefore, it is now possible – in every intermediate city along an indirect service – to make the distinction between a stop and a connection (change of vehicle). Such a distinction is not possible at level 1 when an indirect service consists of direct services of the same type.

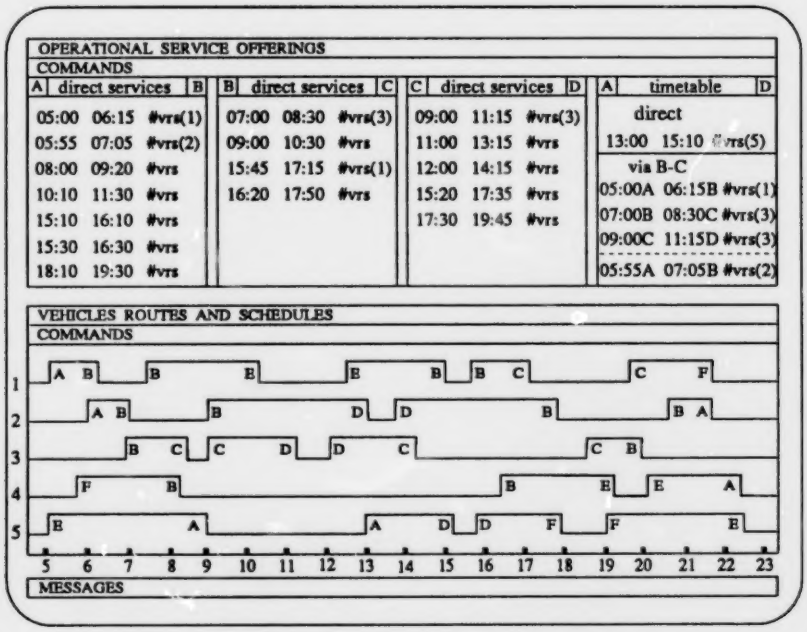

Fig. 4. Main user interface at level 2.

A secondary user interface – for consultation only – has been implemented at level 2 and is shown in Fig. 5. This provides the user with a space-time representation of the vehicle operations currently displayed in Fig. 4. The vertical axis identifies the cities, while the horizontal axis represents the time scale. Each line within the graph represents a vehicle route and schedule, with each horizontal section corresponding to a stop in a city and each oblique line corresponding to a movement. Each such vehicle route has its own specific colour, which is also the colour of its identification number (#vrs) shown in the upper left hand corner of the screen. The main utility of this graphic is to bring out any inconsistencies in the vehicle operations and to highlight the vehicles which are available in a given city at a given time as well as the route sections which are common or almost common to different vehicles, thus suggesting different manipulations to the user. Note that when the graphic is covered with too many routes, the user can delete any of these.

3.3 Consolidation.

The *Consolidation Module* for comparing the market-oriented timetable and the operational timetable is composed of a simple algorithm which generates the union between the two sets of direct services respectively created at levels 1 and 2. The result of this union appears upon request in the upper half of the main user interface at level 2 (Fig. 4), with the services common to both levels in one colour, the services belonging only to level 1 in another colour, and the services belonging only to level 2 in a third colour.

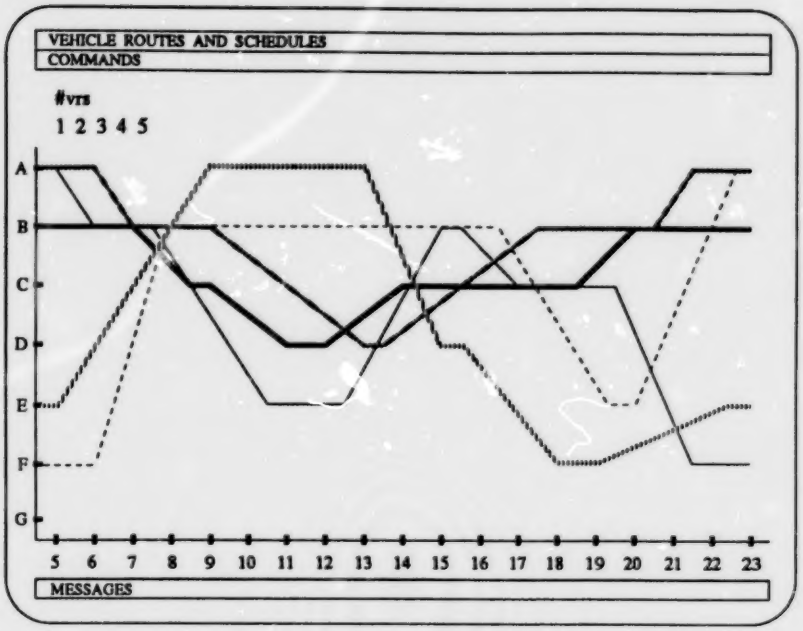

Fig. 5. Vehicle operations in space and time.

Presently, a service is considered common to both levels if it is "identically included" in both (same arrival and departure times, and perfect conformity between the service type at level 1 and the service type corresponding to the vehicle identification number at level 2). However, it would be quite easy to relax this condition of "identical inclusion" so as to disregard minor differences between timetables otherwise identical.

4. DATABASE STRUCTURE AND IMPLEMENTATION EXPERIENCE

The system described in this paper was designed and implemented on an IBM PC/AT (running under DOS), with one major objective: to achieve maximum user-

friendliness, efficiency and speed in entering, consulting, modifying and updating the data relevant to each user interface. More specifically, the system was designed to ensure:

- easy and rapid modification of
  - direct services (level 1)
  - vehicle routes and schedules (level 2)
- instantaneous updating of direct services between all city pairs in response to modifications in vehicle routes and schedules (level 2)
- instantaneous updating of the timetable in response to
  - modifications in direct services (level 1)
  - updating of direct services (level 2).

The database structure developed to achieve these results consists of a *multilist organization* [4], where pointers link the data records (direct and indirect services) in the form of various lists corresponding to the information entities which must be frequently accessed and updated, e.g. vehicle routes and schedules, sets of direct services between all city pairs, timetables in each market, etc. The absence of redundancy in the database (which ensures its integrity and efficient use of the computer memory), direct access to all lists of information in which any service is involved, and the dynamic nature of the list structure are the main advantages of this multilist organization, which is presented in some detail in Appendix 2.

The system has been tested on both a small network and a large one. The corresponding data-sets are outlined in Table 1 which indicates the number of cities, markets, etc. involved in each of the two cases. The first data-set is completely fictitious. The other is based on the national and international flight schedule

operated by Air Canada on a typical weekday in 1976. In both cases, the first four numbers correspond to the data entered into the system, while the two other numbers correspond to the indirect service offerings generated by the system (at level 2) through one and two intermediate cities respectively. The minimum stopping and connecting times were set equal to 20 and 30 minutes respectively; the corresponding maxima were set at 3 hours. Each database was wholly stored in RAM memory although a large number of pointers were involved (around 5,000 and 40,000 pointers for the small and large networks respectively).

In both cases, various types of modifications were repeatedly tested, e.g. inserting a new vehicle route and schedule, eliminating an existing one, rerouting a vehicle through different cities along some part of its initial route, delaying or advancing its departure time from a city, etc. On the average the response of the system was instantaneous, and at the very worst we experienced a response time of about 2 seconds.

Table 1. Data sets.

	Cities	Markets	VRS*	Direct services	Indirect services through 1 city	Indirect services through 2 cities
Small network	7	36	51	188	143	80
Large network	61	550	153	540	1253	669

*Vehicle routes and schedules

5. CONCLUSIONS AND FINAL REMARKS

In this paper, we have presented a micro-computer interactive graphic approach for the integrated design of intercity passenger transportation timetables and corresponding vehicle routes and schedules. The approach provides the ability to deal separately with marketing issues on the one hand and operational issues on the other, but also to reconcile both levels of concern. In addition, the system presented here is as much intended for planning as it is for the "real-time" management of changes in vehicle operations. Our tests indicate that it can be very effective.

In fact, this system was initially developed [5] as part of a new version of a simulation and training game currently under development. (The first version, which does not handle routing and scheduling problems, is described in [6].) However, the approach turns out to be general and flexible enough to be applied and adapted to various real situations.

It is worth noting that the pairing algorithm developed to build the indirect services (made up of two or three direct services, through one of two intermediate cities) could also be used to generate indirect services through more than two cities. In fact, the 1– or 2–stop indirect services could be transferred from the fourth window in the top part of Fig. 4 to the first three windows. The pairing algorithm could then be applied to these indirect services. This would create a timetable where some service offerings will be displayed as a 2– or 3–line entry as presently, but with each line representing a 1– or 2–stop route section. Such a timetable would be quite similar to the information listed in the Official Airline Guide, for example.

Acknowledgement

This research was supported by the Natural Sciences and Engineering Research Council of Canada and F.C.A.R. of Quebec.

REFERENCES

1. C. Lardinois, Supply models for intercity passenger transport: a review. *Transport Reviews* **7**, 119–143 (1987).
2. L. Bodin, B. Golden, A. Assad and M. Ball, Routing and scheduling of vehicles and crews: the state of the art. *Comput. Opns Res.* **10**, 63–211 (1983).
3. F. Soumis, J.A. Ferland and J.-M. Rousseau, MAPUM: a model for assigning passengers to a flight schedule. *Transpn Res.* **15A**, 155–162 (1981).
4. B.G. Claybrook, *File Management Techniques*. John Wiley and Sons (1983).
5. J. Dion, Contributions au développement d'un simulateur interactif-graphique d'entreprises de transport interurbain de passagers. Master thesis, publication no. 646, Centre de recherche sur les transports, Université de Montréal (1989).
6. C. Lardinois, Simulation, gaming and training in a competitive, multimodal, multicompany intercity passenger transportation environment. *J. Opl Res. Soc.* **40**, 849–861 (1989).

APPENDIX 1

Building and Updating Indirect Services

In this Appendix, we outline the procedure for building and updating the indirect services (or indirect travel alternatives), that is, the pairings of direct services in accordance with the concept of coupled time-segments introduced in Section 3.1. This procedure is used concurrently with the construction or modification of the direct services themselves.

First, some basic terms are introduced or defined more precisely. Consider three cities X, Y, Z. A direct service s(Y,Z) is said to be a "valid" successor of a direct service s(X,Y) if (X,Y,Z) is a plausible itinerary and the difference between the departure time of s(Y,Z) from Y and the arrival time of s(X,Y) at Y is greater than a given minimum inter-service pairing-time as well as smaller than a given maximum inter-service pairing-time. These two constraints correspond to the minimum and maximum stopping and connecting times that must be taken into account at the intermediate city Y. Conversely, s(X,Y) is then said to be a "valid" predecessor of s(Y,Z). An "arrival time-segment" A[s(X,Y) | s*(Y,Z)] at Y is a set of incoming direct services (from X to Y) which have the same outgoing service s*(Y,Z) as their "best" valid successor: there is no other valid successor s(Y,Z) that leaves Y earlier. Conversely, one defines a "departure time-segment" D[s(Y,Z) | s*(X,Y)] at Y as a set of outgoing services (from Y to Z) which have the same incoming service s*(X,Y) as their "best" valid predecessor: there is no other valid predecessor s(X,Y) that reaches Y earlier. The concept of coupled time-segments simply states that, for every plausible

itinerary (X,Y,Z), each service of an arrival time-segment A[s(X,Y) | s*(Y,Z)] must be paired with each service of a departure time-segment D[s(Y,Z) | s*(X,Y)] if and only if $s^*(Y,Z) \in D[s(Y,Z) \mid s^*(X,Y)]$ and $s^*(X,Y) \in A[s(X,Y) \mid s^*(Y,Z)]$.

Due to the minimum pairing-time constraint, some incoming services can arrive at the intermediate city Y within a departure time-segment (this was illustrated in Fig. 3), and some outgoing services can leave Y within an arrival time-segment. Thus, the arrival and departure time-segments can overlap in a rather complex way. Strictly speaking, these time-segments are not really built as such, but the concepts and conditions defined above are used implicitly in the course of a dynamic process where pairings of direct services are formed or cancelled as soon as any such service is created, modified or eliminated.

(A) Insertion of a new service s(X,Y)

Step 1. s(X,Y) is inserted in the list L_X (arr., dep.) of all arrivals and departures at city X, i.e. the chronologically ordered list of the incoming and outgoing direct services at X. s(X,Y) is inserted in a similar list L_Y (arr., dep.) at city Y.

Step 2. The list L_X (arr., dep.) is explored forward and backward from s(X,Y). If s(X,Y) is such that an arrival-time segment at city X has been broken into two new such segments, all existing pairings between the incoming services of the first new segment [i.e. the one before the departure time of s(X,Y)] and services from X to Y are cancelled. This is done for all arrival time-

segments broken by s(X,Y). A similar exploration of the list L_Y (arr., dep.) is carried out at city Y, along with similar cancellations for all departure time-segments broken by s(X,Y).

Step 3. The pairing algorithm (described below) is applied to s(X,Y).

(B) Elimination of an existing service s(X,Y)

Step 1. If, at city X, s(X,Y) is the only valid successor of an arrival time-segment, the first part of the pairing algorithm (described below) is applied to all services of this time-segment, while disregarding s(X,Y) and all successors that do not lead to city Y. This is done for all arrival time-segments for which s(X,Y) is the only valid successor. A similar process is applied at city Y for all departure time-segments for which s(X,Y) is the only valid predecessor. (The second part of the algorithm is used in this case.)

Step 2. All existing pairings that involve s(X,Y) are cancelled.

Step 3. s(X,Y) is removed from the lists L_X (arr., dep.) and L_Y (arr., dep.).

(C) Modification of an existing service s(X,Y)

This is treated as an elimination followed by an insertion.

We now describe the pairing algorithm itself, i.e. the procedure for pairing any given service s(I,J) with the appropriate services departing from J towards any other city and with the appropriate services arriving at I from any other city.

Part 1. Pairings of s(I,J) at city J

The list L_J (arr., dep.) of all arrivals and departures at city J is traversed in time increasing order, starting from s(I,J). In the course of this, various incoming and outgoing services will be encountered which correspond to different travel itineraries with J as an intermediate city. To make the presentation simpler, we deal below with a particular plausible itinerary (I,J,K). Thus, when we speak of arrivals (or arrival time-segments) and successors, we refer to services from I to J and from J to K respectively. All such plausible itineraries are in fact treated concurrently within the same process.

Step 1. A pairing is created with every encountered valid successor until the beginning – denoted by b(I,J) – of the arrival time-segment immediately following the arrival time-segment that includes s(I,J) is met. This occurs in two cases:

(a) s(I,J) has not yet been paired with any successor, and a valid successor is encountered which is not paired with one or more arrivals found after s(I,J) because of the minimum pairing time-constraint. The first of these arrivals is b(I,J).

(b) s(I,J) has been paired with at least one valid successor, and an arrival is encountered. The latter is b(I,J).

Step 2. The traversing of the list L_J (arr., dep.) goes on. A pairing is created with every valid successor encountered [including the one encountered in case (a) above] which is not paired with b(I,J) because of the minimum pairing time-constraint.

The traversing stops when (in Step 1 or Step 2) valid successors are no longer encountered, or when (in Step 2) a valid successor which is paired with b(I,J) is found.

Part 2. Pairings of s(I,J) at city I

The procedure is symmetrical to the one described in Part 1 above: the list L_I (arr., dep.) is traversed in time decreasing order, starting from s(I,J).

The presentation above deals with the indirect services that consist of two direct services. Those consisting of three direct services (i.e. those with two intermediate cities) are created during the same process, but no other pairings need to be generated. Each such indirect service $(s_1 - s_2 - s_3)$ is created as soon as the indirect services $(s_1 - s_2)$ and $(s_2 - s_3)$ exist, provided that $(s_1 - s_2 - s_3)$ corresponds to a plausible travel itinerary.

APPENDIX 2

Database Structure

The database structure implemented for storing and processing data is based on a "multilist organization" where data records (each representing a direct or indirect service) are linked in different lists corresponding to the information entities handled by the system, e.g. the vehicle routes and schedules, the timetable for every city pair, etc. Thus, every data record can be a member of various lists. Each such record consists of two parts. One part contains the relevant attributes of the service (place and time of departure and arrival, type of service, etc.). The other part is made up of different pointers that constitute the "connections" of the service within the various lists to which it belongs. A simplified representation of this structure is shown in Fig. 6. Its main components are described below.

Vehicle routes and schedules

- All direct services provided by a specific vehicle are joined in a list that is chronologically ordered by the departure times of the services. The list is doubly-linked: each of its elements is assigned two pointers, one pointing to the previous element in the list, the other pointing to the next. Each such list corresponds to a vehicle route and schedule and can be accessed with a vehicle index (vector of vehicles).

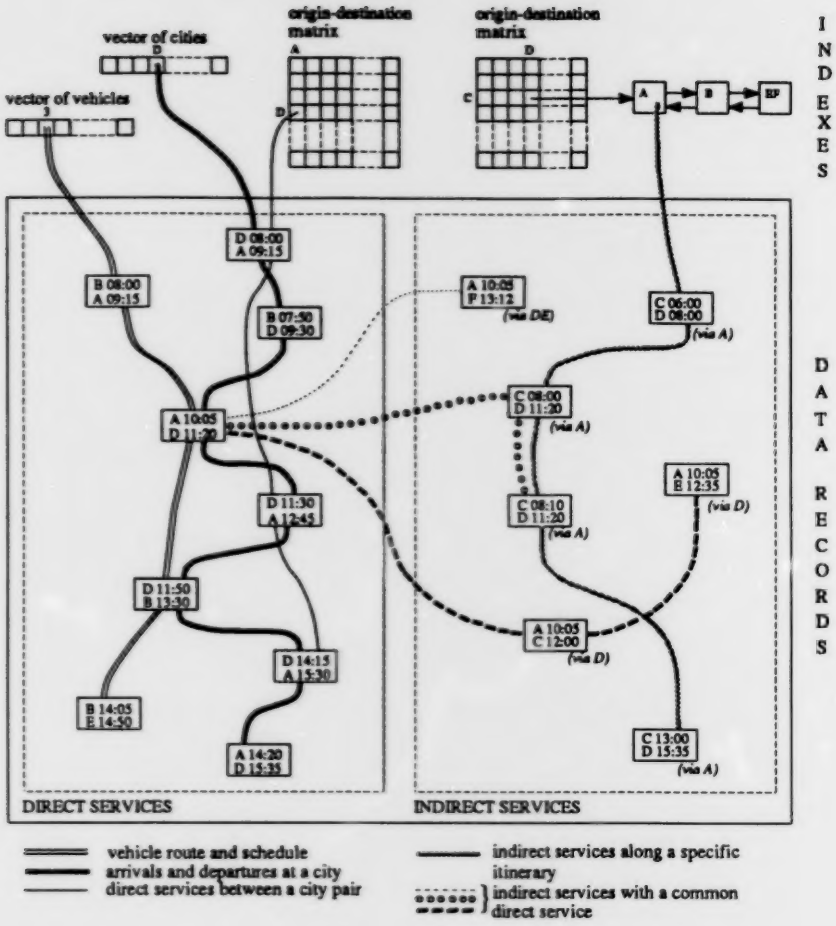

Fig. 6. Multilist organization of the database

Timetable

- All direct services between a specific city pair are joined in a doubly-linked list that is chronologically ordered by the departure times of the services. An origin-destination index (matrix) gives access to every such list.
- All indirect services following the same itinerary are joined in a doubly-linked list that is chronologically ordered by the departure times of the services. An

itinerary index (defined by an origin-destination matrix associated with the lists of intermediate cities for all origin-destination pairs) gives access to every such list.

Arrivals and departures at a city

- All direct services that arrive at and depart from a specific city are joined in a doubly-linked list that is chronologically ordered by the arrival/departure times of the services at this city. These lists are accessed with a city index (vector of cities).

Indirect services having a direct service in common

- All such indirect services are joined in three separate (unordered) doubly-linked lists: one list linking the indirect services made up of two direct services, with the first one as the common service; one list linking the indirect services also made up of two direct services, but with the second one as the common service; one list linking the indirect services made up of three direct services, whatever the location of the one in common. Each direct service thus gives access to all indirect services that use it. (Furthermore, each indirect service has direct access to the two or three direct services composing it; this is not pictured in Fig. 6.)

Finally, note that the overall database consists of two subsets, i.e. one for each of the two levels of decision making extensively discussed in this paper. Both components are organized exactly in the same way as described above [with one exception: at the strategic level (level 1), the vehicles (and hence the corresponding lists defining their routes and schedules) are absent].